

PageRank optimization applied to spam detection

Olivier Fercoq^{*} INRIA Saclay and CMAP Ecole Polytechnique
olivier.fercoq@inria.fr

ABSTRACT

We give a new link spam detection and PageRank demotion algorithm called MaxRank. Like TrustRank and AntiTrustRank, it starts with a seed of hand-picked trusted and spam pages. We define the MaxRank of a page as the frequency of visit of this page by a random surfer minimizing an average cost per time unit. On a given page, the random surfer selects a set of hyperlinks and clicks with uniform probability on any of these hyperlinks. The cost function penalizes spam pages and hyperlink removals. The goal is to determine a hyperlink deletion policy that minimizes this score. The MaxRank is interpreted as a modified PageRank vector, used to sort web pages instead of the usual PageRank vector. The bias vector of this ergodic control problem, which is unique up to an additive constant, is a measure of the “spaminess” of each page, used to detect spam pages. We give a scalable algorithm for MaxRank computation that allowed us to perform experimental results on the WEBSPPAM-UK2007 dataset. We show that our algorithm outperforms both TrustRank and AntiTrustRank for spam and nonspam page detection.

1. INTRODUCTION

Internet search engines use a variety of algorithms to sort web pages based on their text content or on the hyperlink structure of the web. In this paper, we focus on algorithms that use the latter hyperlink structure, called link-based algorithms. The basic notion for all these algorithms is the web graph, which is a digraph with a node for each web page and an arc between pages i and j if there is a hyperlink from page i to page j .

One of the main link-based ranking methods is the PageRank introduced by Brin and Page [10]. It is defined as the invariant measure of a walk made by a random surfer on the web graph. When reading a given page, the surfer

either selects a link from the current page (with a uniform probability), and moves to the page pointed by that link, or interrupts his current search, and then moves to an arbitrary page, which is selected according to given “teleportation” probabilities. The rank of a page is defined as its frequency of visit by the random surfer. It is interpreted as the “popularity” of the page.

From the early days of search engines, some webmasters have tried to get their web pages overranked thanks to malicious manipulations. For instance, adding many keywords on a page is a classical way to make search engines consider a page relevant to many queries. With the advent of link-based algorithms, spammers have developed new strategies, called link-spamming [19], that intend to give some target page a high score. For instance, Gyöngyi and Garcia-Molina [18] showed various linking strategies that improve the PageRank score of a page. They justified the presence of link farms with patterns with every page linking to one single page. Baeza-Yates, Castillo and López [5] also showed that making collusions is a good way to improve PageRank.

In order to fight such malicious manipulations that deteriorate search engines’ results and deceive web surfers, various techniques have been developed. We refer to [11] for a detailed survey of this subject. Each spam detection algorithm is focused on a particular aspect of spam pages. Content analysis (see [25] for instance) is the main tool to detect deceiving keywords. Some simple heuristics [2] may be enough to detect the most coarse link-spam techniques, but more evolved graph algorithms like clique detection [27], SpamRank [8] or Truncated PageRank [7] have also been developed to fight link-spamming. As web spammers adapt themselves to detection algorithms, machine learning techniques [16] try to discover actual predominant spam strategies and to adapt to its evolutions. Another direction of research concerns the propagation of trust through the web graph with the TrustRank algorithm [20] and its variants or the propagation of distrust through a reversed web graph with the AntiTrustRank algorithm [22].

In this paper, we develop a new link spam detection and PageRank demotion algorithm called MaxRank. Like in [20, 22], we start with a seed of hand-picked trusted and spam pages. We define the MaxRank of a page as the frequency of visit of this page by a random surfer minimizing an average cost per time unit. On a given page, the random surfer selects a set of hyperlinks and clicks with uniform probab-

^{*}The doctoral work of the author is supported by Orange Labs through the research contract CRE 3795 with INRIA.

ity on any of these hyperlinks. The cost function penalizes spam pages and hyperlink removals. The goal is to determine an optimal hyperlink deletion policy. The features of MaxRank are based on PageRank optimization [3, 24, 14, 21, 13] and more particularly on the results of [15]. Those papers have shown that the problem of optimizing the PageRank of a set of pages by controlling some hyperlinks can be solved by Markov Decision process algorithms. In these works, the optimization of PageRank was thought from a webmaster's point of view whereas here, we take the search engine's point of view. We show that the Markov Decision Process defining MaxRank is solvable in polynomial time (Proposition 2), because the polytopes of occupation measures admit efficient (polynomial time) separation oracles. The invariant measure of the Markov Decision Process, the MaxRank vector, is interpreted as a modified PageRank vector, used to sort web pages instead of the usual PageRank vector. The solution of the ergodic dynamic programming equation, called the bias vector, is unique up to an additive constant and is interpreted as a measure of the "spaminess" of each page, used to detect spam pages.

We give a scalable algorithm for MaxRank computation that allowed us to perform numerical experiments on the WEBSPPAM-UK2007 dataset [1]. We show that our algorithm outperforms both TrustRank and AntiTrustRank for spam and nonspam page detection. As an example, on the WEBSPPAM-UK2007 dataset, for a recall of 0.8 in the spam detection problem, MaxRank has a precision of 0.87 while TrustRank has a precision of 0.30 and AntiTrustRank a precision of 0.13.

2. THE PAGERANK AND (ANTI)-TRUSTRANK ALGORITHMS

We first recall the basic elements of the Google PageRank computation, see [10] and [23] for more information. We call *web graph* the directed graph with a node per web page and an arc from page i to page j if page i contains a hyperlink to page j . We identify the set of pages to $[n] := \{1, \dots, n\}$.

Let D_i denote the number of hyperlinks contained in page i . Assume first that $D_i \geq 1$ for all $i \in [n]$, meaning that every page has at least one outlink. Then, we construct the $n \times n$ stochastic matrix S , which is such that

$$S_{i,j} = \begin{cases} D_i^{-1} & \text{if page } j \text{ is pointed to from page } i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We also fix a row vector $z \in \mathbb{R}_+^n$, the *zapping or teleportation* vector, which must be stochastic (so, $\sum_{j \in [n]} z_j = 1$), together with a *damping factor* $\alpha \in [0, 1]$ (typically $\alpha = 0.85$) and define the new stochastic matrix

$$P = \alpha S + (1 - \alpha)ez$$

where e is the (column) vector in \mathbb{R}^n with all entries equal to 1.

When some page i has no outlink, $D_i = 0$, and so the entries of the i th row of the matrix S cannot be defined according to (1). Then, we set $S_{i,j} := z_j$.

Consider now a Markov chain $(X_t)_{t \geq 0}$ with transition ma-

trix P , so that for all $i, j \in [n]$, $\mathbb{P}(X_{t+1} = j | X_t = i) = P_{i,j}$. Then, X_t represents the position of a websurfer at time t : when at page i , the websurfer continues his current exploration of the web with probability α and moves to the next page by following the links included in page i , as above, or with probability $1 - \alpha$, stops his current exploration and then teleports to page j with probability z_j .

The *PageRank* π is defined as the invariant measure of the Markov chain $(X_t)_{t \geq 0}$ representing the behavior of the websurfer. This invariant measure is unique if $\alpha < 1$ or if P is irreducible.

The TrustRank algorithm [20] is a semi-automatic algorithm designed for spam detection and PageRank demotion. It starts with a seed of hand-picked trusted pages and then launches the PageRank algorithm with a teleportation vector with nonzero entries only on this seed of trusted pages. Thus the initial trust score of trusted pages will propagate through hyperlinks. The fundamental idea is that trusted pages link to trusted pages and spam pages are linked to by spam pages [12]. Each coordinate of the vector obtained this way gives the TrustRank score of the associated page. Pages with a high TrustRank score are trusted and considered to be non spam, pages with a small TrustRank score are untrusted and considered to be spam. The boundary between trusted and untrusted pages is an arbitrary threshold in TrustRank. As it is a variant of PageRank, the TrustRank score can also be used as such to rank pages instead of the usual PageRank. This is called PageRank demotion, where untrusted pages get a negative promotion of their score.

The AntiTrustRank algorithm [22] follows the same idea as TrustRank but it uses a seed of spam pages instead of a seed of trusted pages and launches the PageRank algorithm using reversed arcs, i.e., it considers a reversed web graph where there is an arc between nodes i and j if page j points to page i . Pages with a high AntiTrustRank are then considered to be spam pages.

3. WELL-DESCRIBED MARKOV DECISION PROCESSES

The definition of the MaxRank relies on the notion of Markov Decision Processes that we recall now.

A finite *Markov decision process* is a 4-uple $(I, (A_i)_{i \in I}, p, c)$ where I is a finite set called the *state space*; for all $i \in I$, A_i is the finite set of *admissible actions* in state i ; $p : I \times \cup_{i \in I} (\{i\} \times A_i) \rightarrow \mathbb{R}_+$ is the *transition law*, so that $p(j|i, a)$ is the probability to go to state j from state i when action $a \in A_i$ is selected; and $c : \cup_{i \in I} (\{i\} \times A_i) \rightarrow \mathbb{R}$ is the *cost function*, so that $c(i, a)$ is the instantaneous cost when action a is selected in state i .

Let $X_t \in I$ denote the state of the system at the discrete time $t \geq 0$. A *deterministic control strategy* ν is a sequence of actions $(\nu_t)_{t \geq 0}$ such that for all $t \geq 0$, ν_t is a function of the history $h_t = (X_0, \nu_0, \dots, X_{t-1}, \nu_{t-1}, X_t)$ and $\nu_t \in A_{X_t}$. Of course, $\mathbb{P}(X_{t+1} = j | X_t, \nu_t) = p(j | X_t, \nu_t), \forall j \in [n], \forall t \geq 0$. More generally, we may consider *randomized strategies* ν where ν_t is a probability measure on A_{X_t} . A strategy ν is *stationary* (feedback) if there exists a function $\bar{\nu}$ such that for all $t \geq 0$, $\nu_t(h_t) = \bar{\nu}(X_t)$.

Given an initial distribution μ representing the law of X_0 , the average cost infinite horizon Markov decision problem, also called *ergodic control problem*, consists in maximizing

$$\liminf_{T \rightarrow +\infty} \frac{1}{T} \mathbb{E} \left(\sum_{t=0}^{T-1} c(X_t, \nu_t) \right) \quad (2)$$

where the maximum is taken over the set of randomized control strategies ν . Indeed, the supremum is the same if it is taken only over the set of randomized (or even deterministic) stationary feedback strategies (Theorem 9.1.8 in [26] for instance).

A Markov decision process is *unichain* if the transition matrix corresponding to every stationary policy has a single recurrent class. Otherwise it is *multichain*. When the problem is unichain, its value does not depend on the initial distribution whereas when it is not, one may consider a vector $(g_i)_{i \in I}$ where g_i represents the value of the problem (2) when starting from state i , meaning that the law of X_0 is the Dirac measure at point i .

We shall need an extension of the formalism of Markov Decision processes in which the actions are implicitly described. This extension, developed in [15], relies on the theory of Groetschel, Lovász and Schrijver [17] of linear programming over polyhedra given by separation oracles.

DEFINITION 1 (DEF. 6.2.2 OF [17]). *We say that a polyhedron \mathcal{B} has facet-complexity at most ϕ if there exists a system of inequalities with rational coefficients that has solution set \mathcal{B} and such that the encoding length of each inequality of the system (the sum of the number of bits of the rational numbers appearing as coefficients in this inequality) is at most ϕ .*

A well-described polyhedron is a triple $(\mathcal{B}; n, \phi)$ where $\mathcal{B} \subseteq \mathbb{R}^n$ is a polyhedron with facet-complexity at most ϕ . The encoding length of \mathcal{B} is by definition $n + \phi$.

DEFINITION 2 (PROB. (2.1.4) OF [17]). *A strong separation oracle for a set K is an algorithm that solves the following problem: given a vector y , decide whether $y \in K$ or not and if not, find a hyperplane that separates y from K ; i.e., find a vector u such that $u^T y > \max\{u^T x, x \in K\}$.*

Inspired by Definition 1, we introduced the following notion.

DEFINITION 3 ([15]). *A finite Markov decision process $(I, (A_i)_{i \in I}, p, c)$ is well-described if for every state $i \in I$, we have $A_i \subset \mathbb{R}^{L_i}$ for some $L_i \in \mathbb{N}$, if there exists $\phi \in \mathbb{N}$ such that the convex hull of every action set A_i is a well-described polyhedron $(\mathcal{B}_i; L_i, \phi)$ with a polynomial time strong separation oracle, and if the costs and transition probabilities satisfy $c(i, a) = \sum_{l \in [L_i]} a_l C_i^l$ and $p(j|i, a) = \sum_{l \in [L_i]} a_l Q_{i,j}^l$, $\forall i, j \in I$, $\forall a \in A_i$, where C_i^l and $Q_{i,j}^l$ are given rational numbers, for $i, j \in I$ and $l \in [L_i]$.*

The encoding length of a well-described Markov decision process is by definition the sum of the encoding lengths of

the rational numbers $Q_{i,j}^l$ and C_i^l and of the well-described polyhedra \mathcal{B}_i .

The interest of well described Markov decision processes is that, even if they have a number of actions exponential in the size of the problem, Theorem 3 in [15] shows that the average cost infinite horizon problem for a well-described (multichain) Markov decision process can be solved in a time polynomial in the input length.

4. THE MAXRANK ALGORITHM

In this section, we define the MaxRank algorithm. It is based on our earlier works on PageRank optimization [15]. In [15], we considered the problem of optimizing the PageRank of a given website from a webmaster's point of view, that is with some controlled hyperlinks and design constraints. Here, we take the search engine's point of view. Hence, for every hyperlink of the web, we can choose to take it into account or not: our goal is to forget spam links while letting trusted links active in the determination of the ranking.

As in TrustRank [20] and AntiTrustRank [22], we start with a seed of trusted pages and known spam pages. The basic idea is to minimize the sum of PageRank scores of spam pages and maximize the sum of PageRank scores of non-spam pages, by allowing us to remove some hyperlinks when computing the PageRank. However, if we do so, the optimal strategy simply consists in isolating spam pages from trusted pages: there is then no hope to detect other spam and nonspam pages. Thus, we add a penalty when a hyperlink is removed, so that "spamcity" can still propagate through (removed or not removed) hyperlinks. Finally, we also control the teleportation vector in order to penalize further pages that we suspect to be spam pages.

We model this by a controlled random walk on the web graph, in which the hyperlinks can be removed. Each time the random surfer goes to a spam page, he gets a positive cost, each time he goes to a trusted page, he gets a negative cost. When the status of the page is unknown, no cost incurs. In addition to this a priori cost, he gets a penalty for each hyperlink removed. Like for PageRank, the random surfer teleports with probability α at every time step; however, in this framework, he chooses the set of pages to which he wants to teleport.

Let \mathcal{F}_x be the set of pages pointed by x in the original graph and D_x be the degree of x . An action consists in determining $J \subseteq \mathcal{F}_x$, the set of hyperlinks that remain, and $I \subseteq [n]$, the set of pages to which the surfer may teleport. We shall restrict I to have a cardinality equal to $N \leq n$. Then, the probability of transition from page x to page y is

$$p(y|x, I, J) = \alpha \nu_y(I, J) + (1 - \alpha) z_y(I)$$

where the teleportation vector and the hyperlink click probability distribution are given by

$$z_y(I) = \begin{cases} |I|^{-1} & \text{if } y \in I \\ 0 & \text{otherwise} \end{cases}$$

$$\nu_y(I, J) = \begin{cases} z_y(I) & \text{if } |J| = \emptyset \\ |J|^{-1} & \text{if } y \in J \\ 0 & \text{otherwise} \end{cases}$$

The cost at page x is given by

$$c(x, I, J) = c'_x + \gamma \frac{D_x - |J|}{D_x} .$$

c'_x is the a priori cost of page x . This a priori cost should be positive for known spam pages and negative for trusted pages. D_x is the degree of x in the web graph and $\gamma > 0$ is a penalty factor. The penalty $\gamma \frac{D_x - |J|}{D_x}$ is proportional to the number of pages removed.

We study the following ergodic control problem:

$$\inf_{(I_t)_{t \geq 0}, (J_t)_{t \geq 0}} \limsup_{T \rightarrow +\infty} \frac{1}{T} \mathbb{E} \left(\sum_{t=0}^{T-1} c(X_t, I_t, J_t) \right) , \quad (3)$$

where an admissible control consists in selecting, at each time step t , a subset of pages $I_t \subseteq [n]$ with $|I_t| = N$ to which teleportation is permitted, and a subset $J_t \subseteq \mathcal{F}_{X_t}$ of the set of hyperlinks in the currently visited page X_t .

The following proposition gives an alternative formulation of Problem (3) that we will then show to be well-described.

PROPOSITION 1. Fix $N \in \mathbb{N}$ and let

$$Z = \{z \in \mathbb{R}^n \mid \sum_{i \in [n]} z_i = 1, 0 \leq z_i \leq \frac{1}{N}\} .$$

Let \mathcal{F}_x be the set of pages pointed by x in the original graph and D_x be the degree of x . Let \mathcal{P}_x be the polyhedron defined as the set of vectors $(\sigma, \nu) \in \mathbb{R}^{D_x+1} \times \mathbb{R}^n$ such that there exists $w \in \mathbb{R}^{(D_x+1) \times n}$ verifying

$$\sum_{d=0}^{D_x} \sigma^d = 1 \quad (4a)$$

$$\sigma^d \geq 0, \quad \forall d \in \{0, \dots, D_x\} \quad (4b)$$

$$\nu_j = \sum_{d=0}^{D_x} w_j^d, \quad \forall j \in [n] \quad (4c)$$

$$\sum_{j \in [n]} w_j^d = \sigma^d, \quad \forall d \in \{0, \dots, D_x\} \quad (4d)$$

$$0 \leq w_j^0 \leq \frac{\sigma^0}{N_z}, \quad \forall j \in [n] \quad (4e)$$

$$w_j^d = 0, \quad \forall j \notin \mathcal{F}_x, \forall d \in \{1, \dots, D_x\} \quad (4f)$$

$$0 \leq w_j^d \leq \frac{\sigma^d}{d}, \quad \forall j \in \mathcal{F}_x, \forall d \in \{1, \dots, D_x\} \quad (4g)$$

Then Problem 3 is equivalent to the following ergodic control problem

$$\inf_{\sigma, \nu, z} \limsup_{T \rightarrow +\infty} \frac{1}{T} \mathbb{E} \left(\sum_{t=0}^{T-1} \tilde{c}(X_t, \sigma_t, \nu_t, z_t) \right) , \quad (5)$$

where the cost is defined as

$$\tilde{c}(x, \sigma, \nu, z) = c'_x + \gamma \frac{D_x - \sum_{d=0}^{D_x} d \sigma^d}{D_x}$$

and the transitions are

$$\tilde{p}(y|x, \sigma, \nu, z) = \alpha \nu_y + (1 - \alpha) z_y .$$

The admissible controls verify for all t , $(\sigma_t, \nu_t) \in \text{extr}(\mathcal{P}_{X_t})$ (the set of extreme point of the polytope) and $z_t \in \text{extr}(Z)$.

Indeed, to each action (σ, ν, z) of Problem (5) corresponds a unique action I, J of Problem (3) and vice versa. Moreover, the respective transitions and costs are equal.

PROOF. Fix a page x in $[n]$. The extreme points of Z are the vectors of \mathbb{R}^n with N coordinates equal to $\frac{1}{N}$ and the other ones equal to 0. Hence $z(I) \in \text{extr}(Z)$ and for each extreme point z' of Z there exists $I \in [n]$ such that $|I| = N$ and $z' = z(I)$. We shall also describe the set of extreme points of \mathcal{P}_x .

From the theory of disjunctive linear programming [6], we can see that the polytope $\mathcal{K} = \{(\sigma, \nu) \in \mathcal{P}_x\}$ is the convex hull of the union of $D_x + 1$ polytopes that we will denote K_d , $d \in \{0, 1, \dots, D_x\}$. If $d = 0$, then $K_0 = Z$. If $d > 0$, $K_d = \{(\sigma, \nu) \in \mathbb{R}^n \mid \sum_{j \in [n]} \nu_j = 1, 0 \leq \nu_j \leq \frac{1}{d}, \forall j \in \mathcal{F}_x, \nu_j = 0, \forall j \notin \mathcal{F}_x\}$.

Let (σ, ν) be an extreme point of \mathcal{P}_x . By Corollary 2.1.2-ii) in [6], there exists d^* such that $\sigma^{d^*} = 1$ and ν is an extreme point of \mathcal{K} . As $\sigma^{d^*} = 1$ and $\sigma_d = 0$ for $d \neq d^*$, we conclude that ν is also an extreme point of $K_{d^*} = \mathcal{P}_x \cap \{\sigma \mid \sigma^{d^*} = 1\}$. If $d^* = 0$, $\nu \in \text{extr}(Z)$ and if $d^* > 0$, the extreme points of K_{d^*} correspond exactly to the vectors with d^* coordinates in \mathcal{F}_x equal to $\frac{1}{d^*}$ and the other ones equal to 0. Hence there exists $I \subseteq [n]$ and $J \subseteq \mathcal{F}_x$ such that $|I| = N$, $|J| = d^*$ and $\nu = \nu(I, J)$.

Conversely, fix I and J and let $\sigma(J)$ be such that $\sigma^d(J) = 1$ if and only if $d = |J|$. Then $(\sigma(J), \nu(I, J))$ is an extreme point of \mathcal{P}_x by Corollary 2.1.2-i) in [6].

Finally the costs are the same since $|J| = \sum_{d=0}^{D_x} d \sigma^d$. \square

PROPOSITION 2. If α, γ and $c'_i, i \in [n]$ are rational numbers, then the ergodic control problem (5) is the average cost infinite horizon problem for a well described Markov decision process and it is polynomial time solvable.

PROOF. Clearly, the process described is a Markov decision process. As the polytopes $\mathcal{P}_i, i \in [n]$ and Z are described by a polynomial number of inequalities with at most $n + 1$ terms in each, they are well described. Indeed, the separation oracle consisting simply in testing each inequality terminates in polynomial time. The cost and transitions are linear functions on those polytopes with rational coefficients since α and $c'_i, i \in [n]$ are rational numbers. Thus the Markov decision process is well described. By Theorem 3 in [15], Problem (5) is thus solvable in polynomial time. \square

PROPOSITION 3. *The dynamic programming equation*

$$v_i + \lambda = \min_{(\sigma, \nu) \in \mathcal{P}_i, z \in Z} c'_i + \gamma \frac{D_i - \sum_{d=0}^{D_i} d\sigma^d}{D_i} + \sum_{j \in [n]} (\alpha \nu_j + (1 - \alpha) z_j) v_j, \quad \forall i \in [n] \quad (6)$$

has a solution $v \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. The constant λ is unique and is the value of problem (3). An optimal strategy is obtained by selecting for each state i , $(\sigma, \nu) \in \mathcal{P}_i$ and $z \in Z$ maximizing Equation (6). The function v is called the bias.

PROOF. Theorem 8.4.3 in [Put94] applied to the unichain ergodic control problem (5) implies the result of the proposition but with \mathcal{P}_i replaced by $\text{extr}(\mathcal{P}_i)$. But as the expression which is maximized is affine, using \mathcal{P}_i or $\text{extr}(\mathcal{P}_i)$ yields the same solution. Proposition 1 gives the equivalence between (3) and (5) \square

PROPOSITION 4. *Let T be the dynamic programming operator $\mathbb{R}^n \rightarrow \mathbb{R}^n$ defined by*

$$T_i(v) = \min_{(\sigma, \nu) \in \mathcal{P}_i} c'_i + \gamma \frac{D_i - \sum_{d=0}^{D_i} d\sigma^d}{D_i} + \alpha \sum_{j \in [n]} \nu_j v_j, \quad \forall i \in [n].$$

The map T is α -contracting in the sup norm and its fixed point v , which is unique, is such that $(v, (1 - \alpha) \min_{z \in Z} z \cdot v)$ is solution of the ergodic dynamic programming equation (6).

PROOF. The set $\{\nu \text{ st: } (\sigma, \nu) \in \mathcal{P}_i\}$ is a set of probability measures so $\lambda \in \mathbb{R} \Rightarrow T(v + \lambda) = T(v) + \alpha \lambda$ and $v \geq w \Rightarrow T(v) \geq T(w)$. This implies that T is α -contracting. Let v be its fixed point. For all $i \in [n]$,

$$\begin{aligned} & v_i + (1 - \alpha) \min_{z \in Z} z \cdot v \\ &= \min_{(\sigma, \nu) \in \mathcal{P}_i, z \in Z} c'_i + \gamma \frac{D_i - \sum_{d=0}^{D_i} d\sigma^d}{D_i} + \sum_{j \in [n]} \alpha \nu_j v_j + (1 - \alpha) z_j v_j \end{aligned}$$

We get equation (6) with constant $(1 - \alpha) \min_{z \in Z} z \cdot v$. \square

We can then solve the dynamic programming equation (6) and so the ergodic control problem (3) by value iteration (outer loop of Algorithm 1).

The algorithm starts with an initial potential function v , scans repeatedly the pages and updates v_i when i is the current page according to $v_i \leftarrow T_i(v)$ until convergence is reached. Then $(v, (1 - \alpha) \min_{z \in Z} z v)$ is solution of the ergodic dynamic programming equation (6) and an optimal linkage strategy is recovered by selecting the maximizing (σ, ν) at each page. An optimal teleportation vector is recovered by selecting a maximizing z in $\min_{z \in Z} z v$.

Thanks to the damping factor α , the iteration can be seen to be α -contracting if the pages are scanned in a cyclic order. Thus the algorithm converges in a number of steps independent of the dimension of the web graph.

For the evaluation of the dynamic programming operator T at a page i (inner for-loop of Algorithm 1), we remark

that for a fixed 0 – 1 valued σ , that is for a fixed number of removed hyperlinks, the operator T_i maximizes a linear function on a hypercube, which reduces essentially to a sort. Any extreme point (σ, ν) of \mathcal{P}_i necessarily verifies that σ is 0 – 1 valued. Thus we just need to choose the best value of σ among the $D_i + 1$ possibilities.

Algorithm 1 MaxRank algorithm

```

1: Initialization:  $v \in \mathbb{R}^n$ 
2: while  $\|v - T(v)\|_\infty \geq \epsilon$  do
3:   Sort  $(v_l)_{l \in [n]}$  in increasing order and let  $\phi : [n] \rightarrow [n]$ 
     be the sort function so that  $v_{\phi(1)} \leq \dots \leq v_{\phi(n)}$ .
4:    $\lambda \leftarrow \frac{1-\alpha}{N} \sum_{j=1}^N v_{\phi(j)}$ 
5:   for  $i$  from 1 to  $n$  do
6:      $w_i^0 \leftarrow c'_i + \gamma + \frac{\alpha}{1-\alpha} \lambda$ 
7:     Sort  $(v_j)_{j \in \mathcal{F}_i}$  in increasing order and let  $\psi : \mathcal{F}_i \rightarrow$ 
        $\{1, \dots, |\mathcal{F}_i|\}$  such that  $v_{\psi(1)} \leq \dots \leq v_{\psi(|\mathcal{F}_i|)}$ .
8:     for  $d$  from 1 to  $D_i$  do
9:        $w_i^d \leftarrow c'_i + \gamma \frac{D_i - d}{D_i} + \frac{\alpha}{d} \sum_{j=1}^d v_{\psi(j)}$ 
10:    end for
11:     $v_i \leftarrow T_i(v) = \min_{d \in \{0, 1, \dots, D_i\}} w_i^d$ 
12:  end for
13: end while

```

This very efficient algorithm is highly scalable: we used it for our experimental results on a large size dataset (Section 5).

The following proposition shows that if γ is too big, then the optimal link removal strategy is trivial. It also gives an interpretation of the bias vector in terms of number of visits of spam pages.

PROPOSITION 5. *If $\gamma > \frac{2\alpha}{1-\alpha} \|c\|_\infty$, then no link should be removed. Moreover if in addition, $c'_i = 1$ when i is a spam page and 0 otherwise, then the i th coordinate of the fix point of operator T in Proposition 4 is equal to the expected mean number of spam pages visited before teleportation when starting a walk from page i .*

PROOF. Proposition 4 gives a normalization of the bias vector such that it cannot take any value bigger than $\frac{\|c\|_\infty}{1-\alpha}$, as it is the case for PageRank optimization [14, 15].

Now fix $i \in [n]$. Let $\nu^0 = \nu(\mathcal{F}_i)$ and $\nu = \nu(I, J)$ for $J \subset \mathcal{F}_i$. If $J = \emptyset$, then

$$\alpha |\nu v - \nu^0 v| \leq \alpha |\nu v| + \alpha |\nu^0 v| \leq \frac{2\alpha}{1-\alpha} \|c\|_\infty \frac{D_i - |\emptyset|}{D_i}.$$

If $J \neq \emptyset$, then

$$\begin{aligned} \alpha |\nu v - \nu^0 v| &\leq \alpha \left| \left(\frac{1}{|J|} - \frac{1}{D_i} \right) \sum_{j \in J} v_j \right| + \alpha \left| \frac{1}{D_i} \sum_{j \in \mathcal{F}_i} v_j \right| \\ &\leq \frac{D_i - |J|}{|J| D_i} \frac{\alpha \|c\|_\infty}{1-\alpha} + \frac{D_i - |J|}{D_i} \frac{\alpha \|c\|_\infty}{1-\alpha} \leq \gamma \frac{D_i - |J|}{D_i}. \end{aligned}$$

This proves that choosing $J = \mathcal{F}_i$ is always the best strategy when $\gamma > \frac{2\alpha}{1-\alpha} \|c\|_\infty$.

When no link is removed and c' is defined as in the proposition, we are in the framework of [14], where it is shown that

the i th coordinate of the fix points of the operator T is equal to the expected mean number of visits before teleportation when starting a walk from page i . \square

5. SPAM DETECTION AND PAGERANK DEMOTION

We performed numerical experiments on the WEBSpAM-UK2007 dataset [1]. This is a crawl of the .uk websites with $n = 105,896,555$ pages performed in 2007, associated with lists of hosts classified as spam, nonspam and borderline. There is a training dataset for the setting of the algorithm and a test dataset to test the performance of the algorithm.

We took $\gamma = 4$, $\alpha = 0.85$, $N = 0.89n$, $c'_i = 1$ if i is a spam page of the training dataset, $c'_i = -0.2$ if i is a nonspam page of the training dataset and $c'_i = 0$ otherwise. Then we obtained a MaxRank score and the associated bias vector. We also computed TrustRank and AntiTrustRank with the training dataset as the seed sets. We used the Webgraph framework [9], so that we could manage the computation on a personal computer with four Intel Xeon CPUs at 2.98 Ghz and 8 GB RAM. We coded the algorithm in a parallel fashion thanks to the OpenMP library. Computation took around 6 minutes for each evaluation of the dynamic programming operator of Proposition 4 and 6 hours for 60 such iterations (precision on the objective $\alpha^{60} \leq 6.10^{-5}$). By comparison, PageRank computation with the same precision required 1.3 hour on the same computer, which is of the same order of magnitude.

Figure 1 gives the values taken by the bias vector. Figure 2 compares the precision and recall of PageRank, TrustRank, AntiTrustRank and MaxRank bias for spam or non spam detection. Precision and recall are the usual measures of the quality of an information retrieval algorithm [4]. Precision is the probability that a randomly selected retrieved document is relevant. Recall is the probability that a randomly selected relevant document is retrieved in a search. These values were obtained using the training and the test sets. Figure 3 compares TrustRank and MaxRank scores for PageRank demotion.

6. REFERENCES

- [1] Webspam-uk2007, 2007. Yahoo! Research: "Web Spam Collections". Crawled by the Laboratory of Web Algorithmics, University of Milan.
- [2] T. Abou-Assaleh and T. Das. Combating spamdexing: Incorporating heuristics in link-based ranking. In W. Aiello, A. Broder, J. Janssen, and E. Milios, editors, *Algorithms and Models for the Web-Graph*, volume 4936 of *Lecture Notes in Computer Science*, pages 97–106. Springer Berlin / Heidelberg, 2008.
- [3] K. Avrachenkov and N. Litvak. The Effect of New Links on Google PageRank. *Stochastic Models*, 22(2):319–331, 2006.
- [4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
- [5] R. A. Baeza-Yates, C. Castillo, and V. López. Pagerank increase under different collusion topologies. In *Proc. of the First International Workshop on Adversarial Information Retrieval on the Web*, pages 17–24, 2005.

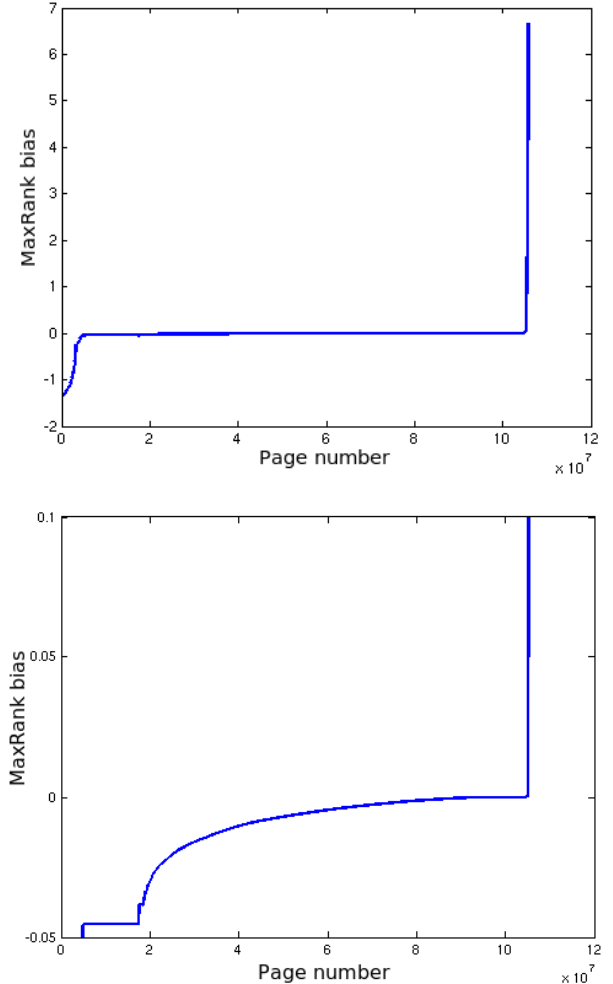


Figure 1: Recognizing spam with the bias vector. Top: the values of the bias for all pages. Bottom: zoom on bias values near 0. Pages are sorted by growing bias value. Spam pages of the training set have a large positive bias value, non spam pages of the training set have a negative bias value. Pages in between describe a “continuum” of values and the separation between pages considered spam or not is arbitrarily set.

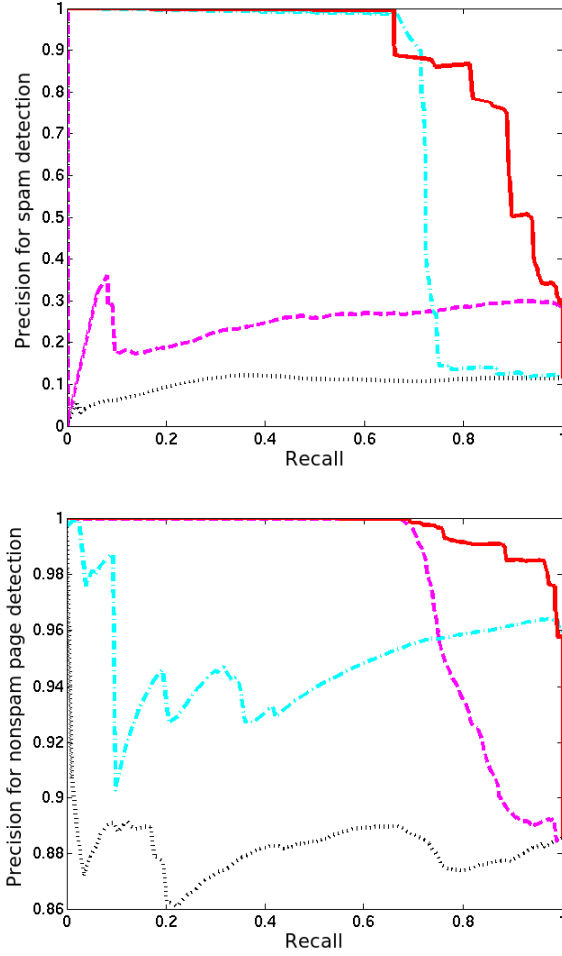


Figure 2: Top: Precision as a function of recall for spam detection. Bottom: Precision as a function of recall for detection of non spam pages. We present the result for four algorithms: PageRank (dotted line below), TrustRank (dashed line), Anti-TrustRank (dash-dotted line) and MaxRank bias (solid line). Given a vector of scores, the various pairs of precision-recall values are obtained for various thresholds defining the discrimination between pages considered spam and non spam. TrustRank and AntiTrustRank have a precision of 1 on their seed set, which represent around 70% of the total test set. But out of their training set, their precision decreases quickly. MaxRank, on the other hand, remains precise even out of its training set.

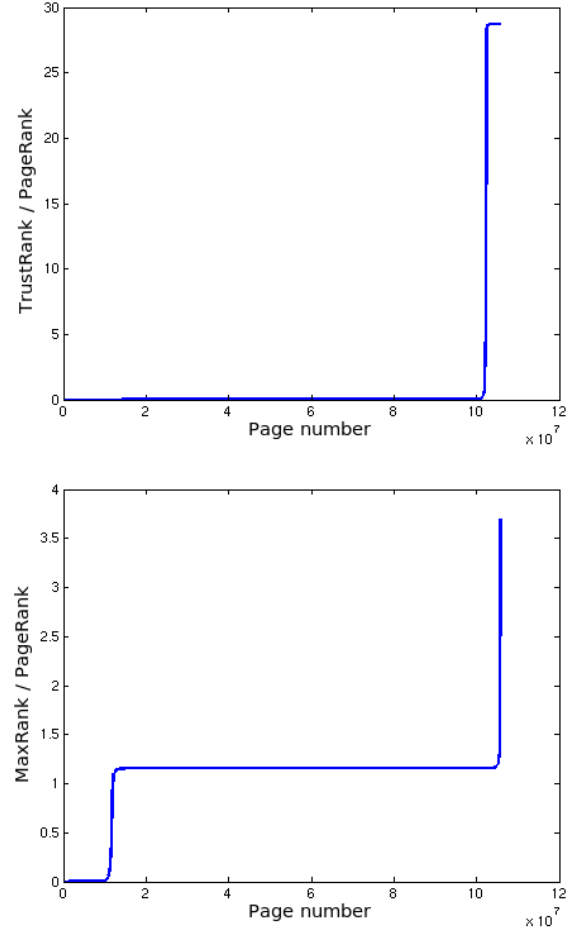


Figure 3: Top: PageRank demotion by TrustRank. We show the ratio of TrustRank over PageRank. Pages are sorted by growing promotion. The demotion is very coarse, since there are many pages with a very small TrustRank score, when compared to their PageRank score. Bottom: PageRank demotion by MaxRank. The demotion is finer, most of the pages have a nearly unchanged score, up to a multiplicative constant.

- [6] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1-3):3–44, 1998.
- [7] L. Becchetti and S. Leonardi. Using rank propagation and probabilistic counting for link-based spam detection. In *Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD)*. ACM Press, 2006.
- [8] A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher. Spamrank – fully automatic link spam detection. In *1st International Workshop on Adversarial Information Retrieval on the Web*, 2005.
- [9] P. Boldi and S. Vigna. The webgraph framework i: Compression techniques. In *Proc. of the Thirteenth International World Wide Web Conference*, pages 595–601. ACM Press, 2004.
- [10] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998. Proc. 17th International World Wide Web Conference.
- [11] C. Castillo and B. Davison. Adversarial web search. In *Foundations and Trends in Information Retrieval*, volume 4, pages 377–486. Now Publishers, 2010.
- [12] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430, New York, 2007. ACM.
- [13] B. C. Csáji, R. M. Jungers, and V. D. Blondel. Pagerank optimization in polynomial time by stochastic shortest path reformulation. In *Proc. 21st International Conference on Algorithmic Learning Theory*, volume 6331 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2010.
- [14] C. de Kerchove, L. Ninove, and P. van Dooren. Maximizing pagerank via outlinks. *Linear Algebra and its Applications*, 429(5-6):1254–1276, 2008.
- [15] O. Fercoq, M. Akian, M. Bouhtou, and S. Gaubert. Ergodic control and polyhedral approaches to Pagerank optimization. *arXiv preprint:1011.2348*, 2010. Conditionally accepted in IEEE TAC.
- [16] Q. Gan and T. Suel. Improving web spam classifiers using link structure. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb ’07, pages 17–20, New York, 2007. ACM.
- [17] M. Groetschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, 1988.
- [18] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *VLDB ’05: Proc. 31st international conference on Very large data bases*, pages 517–528. VLDB Endowment, 2005.
- [19] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005)*, 2005.
- [20] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases*, volume 30, pages 576–587. VLDB Endowment, 2004.
- [21] H. Ishii and R. Tempo. Computing the pagerank variation for fragile web data. *SICE J. of Control, Measurement, and System Integration*, 2(1):1–9, 2009.
- [22] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *Proc. ACM SIGIR workshop on Adversarial Information Retrieval on the Web*, 2006.
- [23] A. N. Langville and C. D. Meyer. *Google’s PageRank and beyond: the science of search engine rankings*. Princeton University Press, 2006.
- [24] F. Mathieu and L. Viennot. Local aspects of the global ranking of web pages. In *6th International Workshop on Innovative Internet Community Systems (I2CS)*, pages 1–10, Neuchâtel, 2006.
- [25] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, pages 83–92, New York, 2006. ACM.
- [26] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
- [27] H. Saito, M. Toyoda, M. Kitsuregawa, and K. Aihara. A large-scale study of link spam detection by graph algorithms. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb ’07, pages 45–48, New York, 2007. ACM.